State of the Art Cocoa

User interface development

Session 426

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

State of the Art Cocoa

User interface development

Joar Wingfors
Ron Lue-Sang

Roadmap

- Getting started
- Cocoa drawing
- Custom controls
- Behaviours

Getting Started

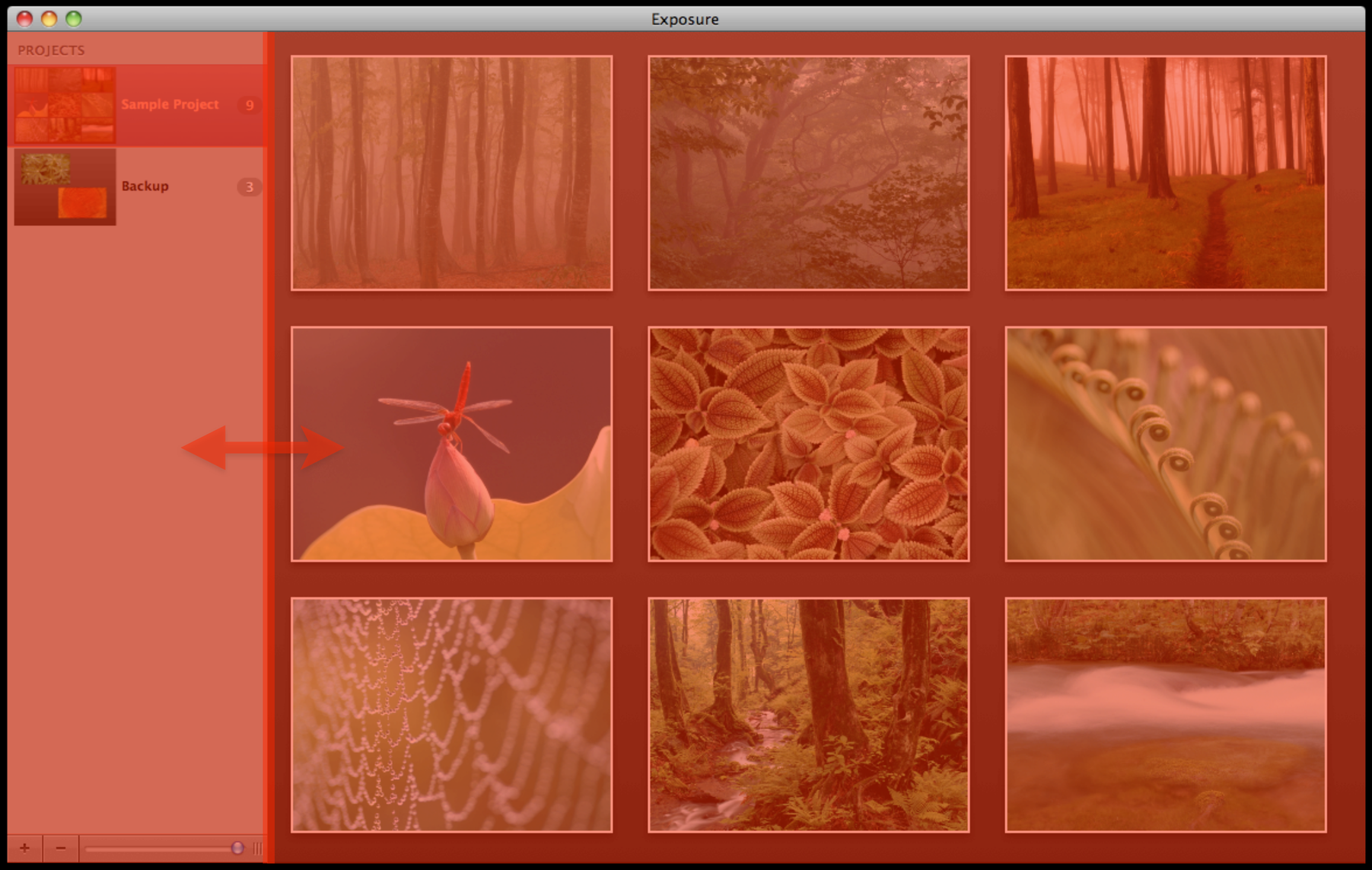
Joar Wingfors
Development Technologies

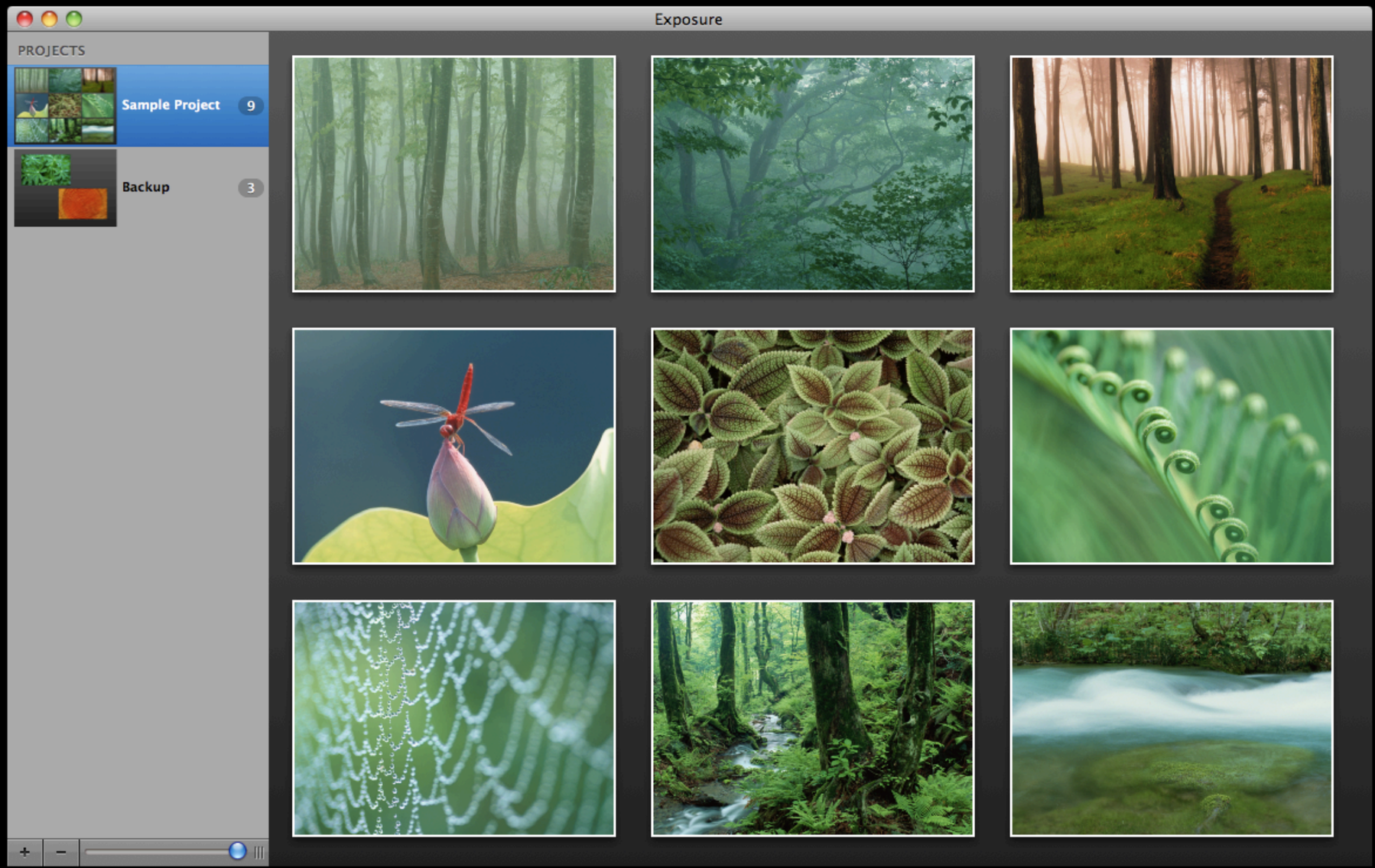
Getting Started

- Integrated process
- Professional UI Designers
- Apple HI Guidelines

Interface Builder

- Visual design tool
- Integrated with Xcode
- Rapid development
- Less code


Exposure
PROJECTS
Sample Project
9
Backup
3

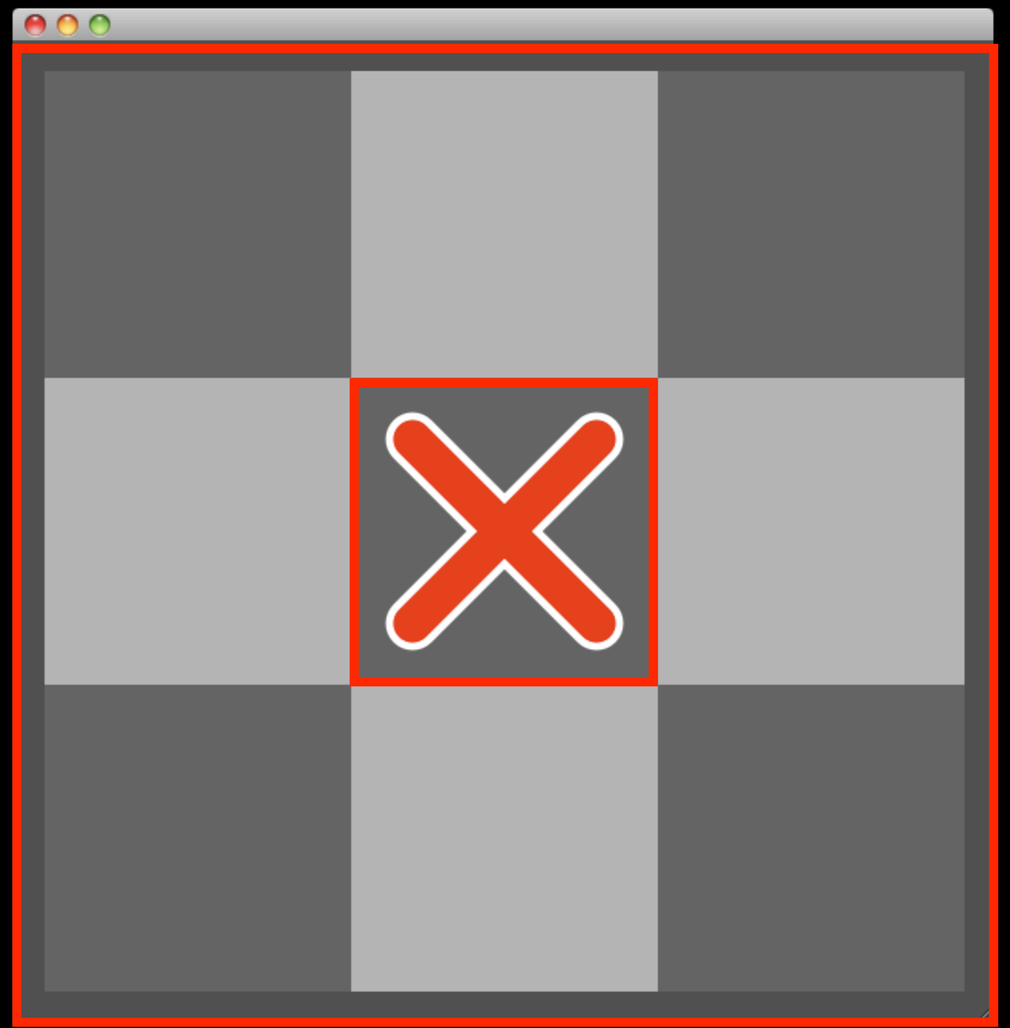

Exposure
PROJECTS
Sample Project
9
Backup
3

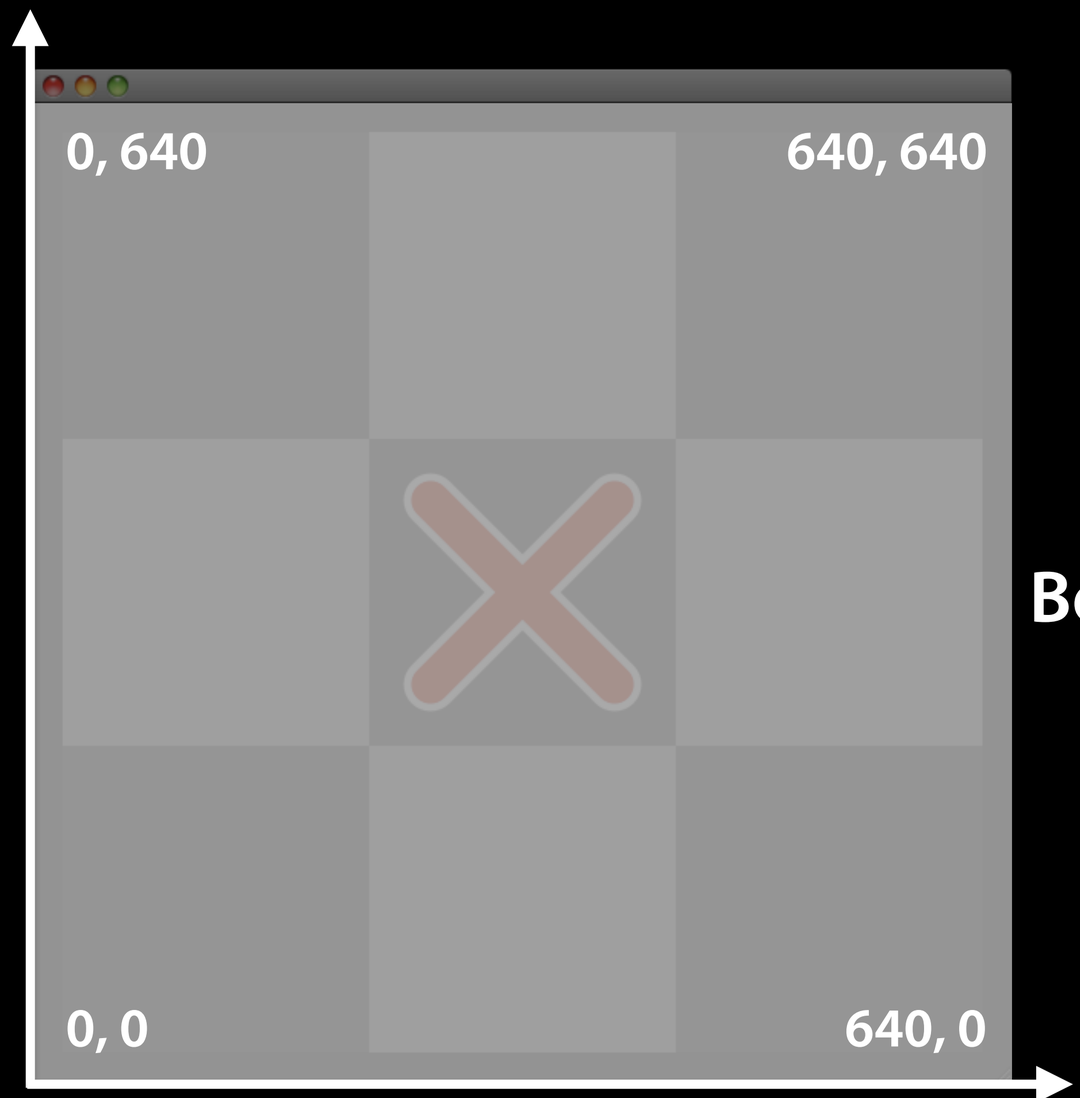
Demo

Cocoa Drawing

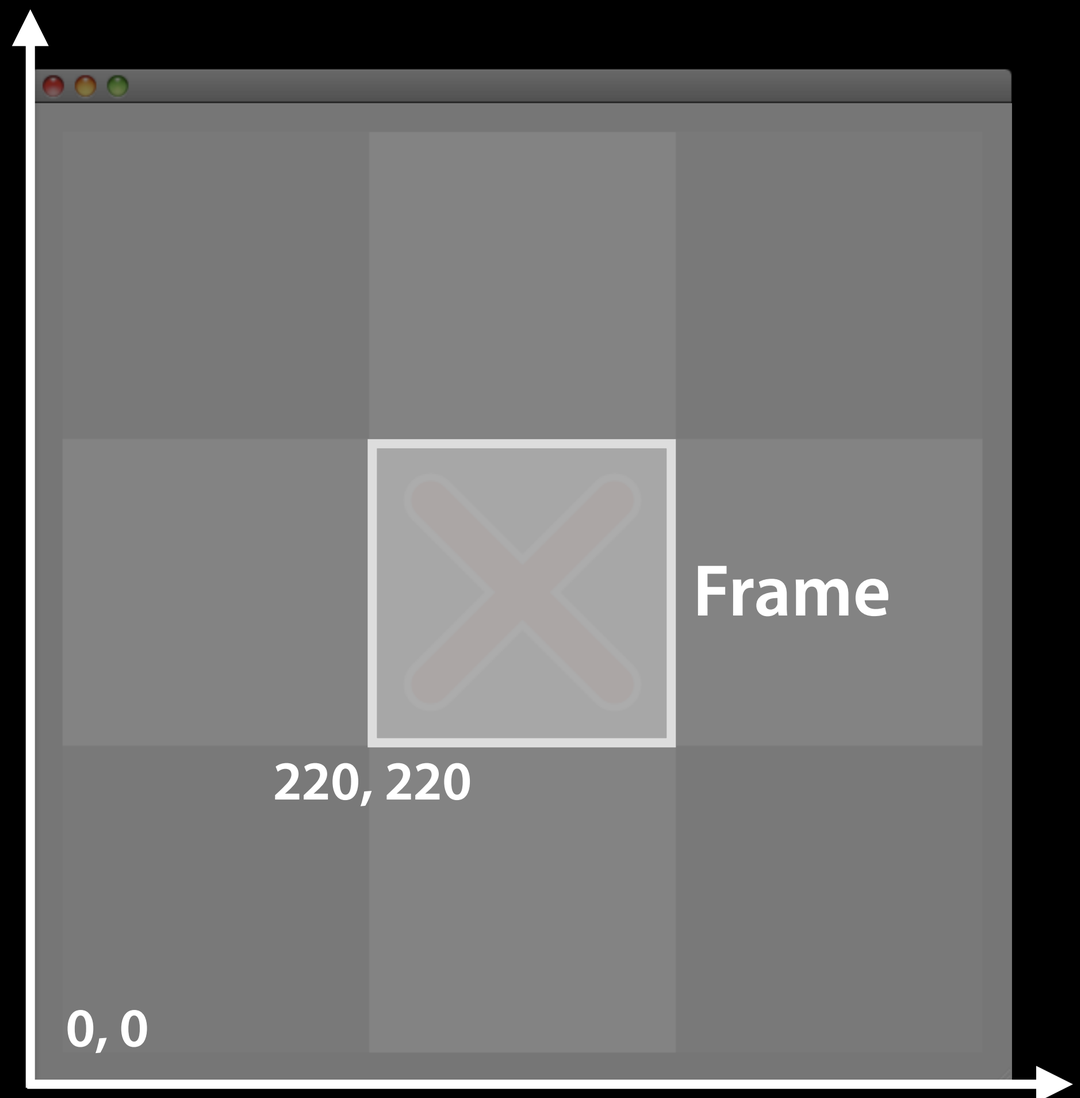
Joar Wingfors
Development Technologies

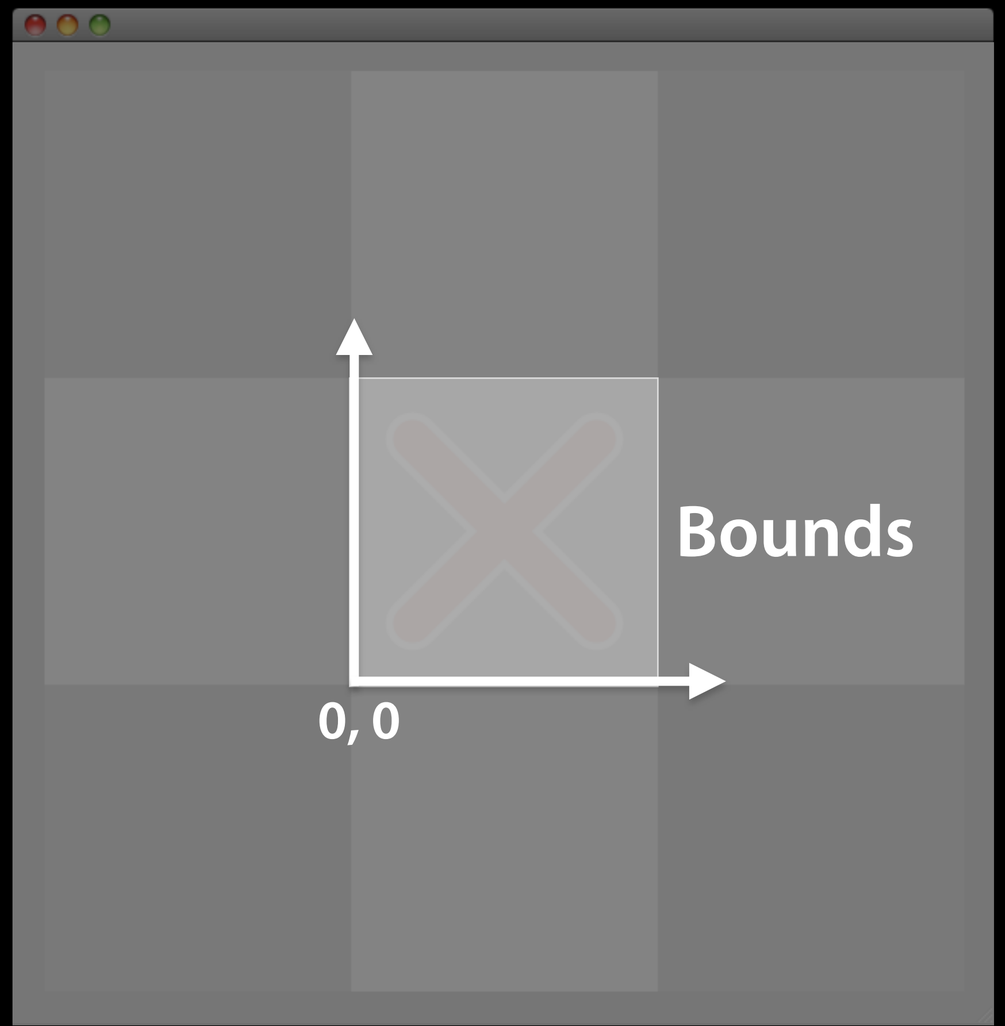
Finder File Edit View Go Window Help
Mon 10:50 AM





Bounds

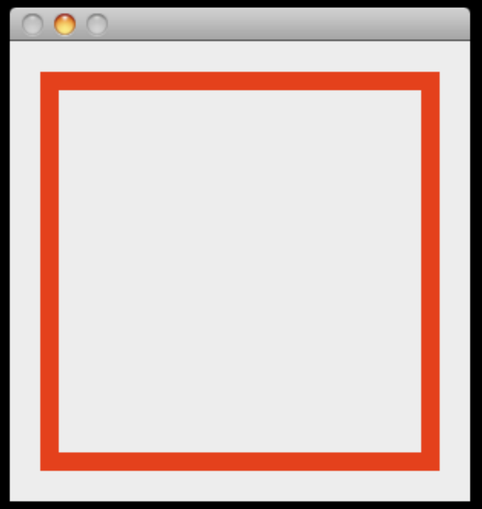

Frame
220, 220
0, 0


Bounds
0, 0

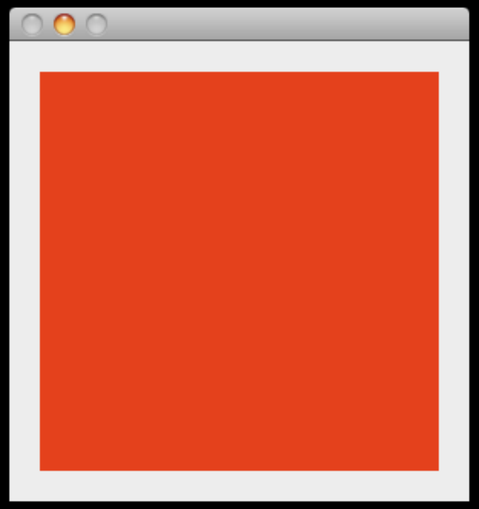
Structs and Functions

- NSPoint, NSSize, NSRect
- NSZeroPoint, NSStringFromPoint(), NSEqualPoints()
- Measurements, Union, Intersect, Inset
- Drawing

NSFrameRect



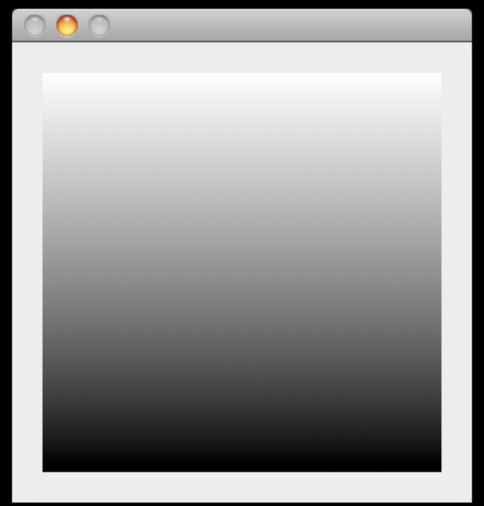
NSRectFill



Object Oriented API

- Gradients
- Paths
- Images
- Text

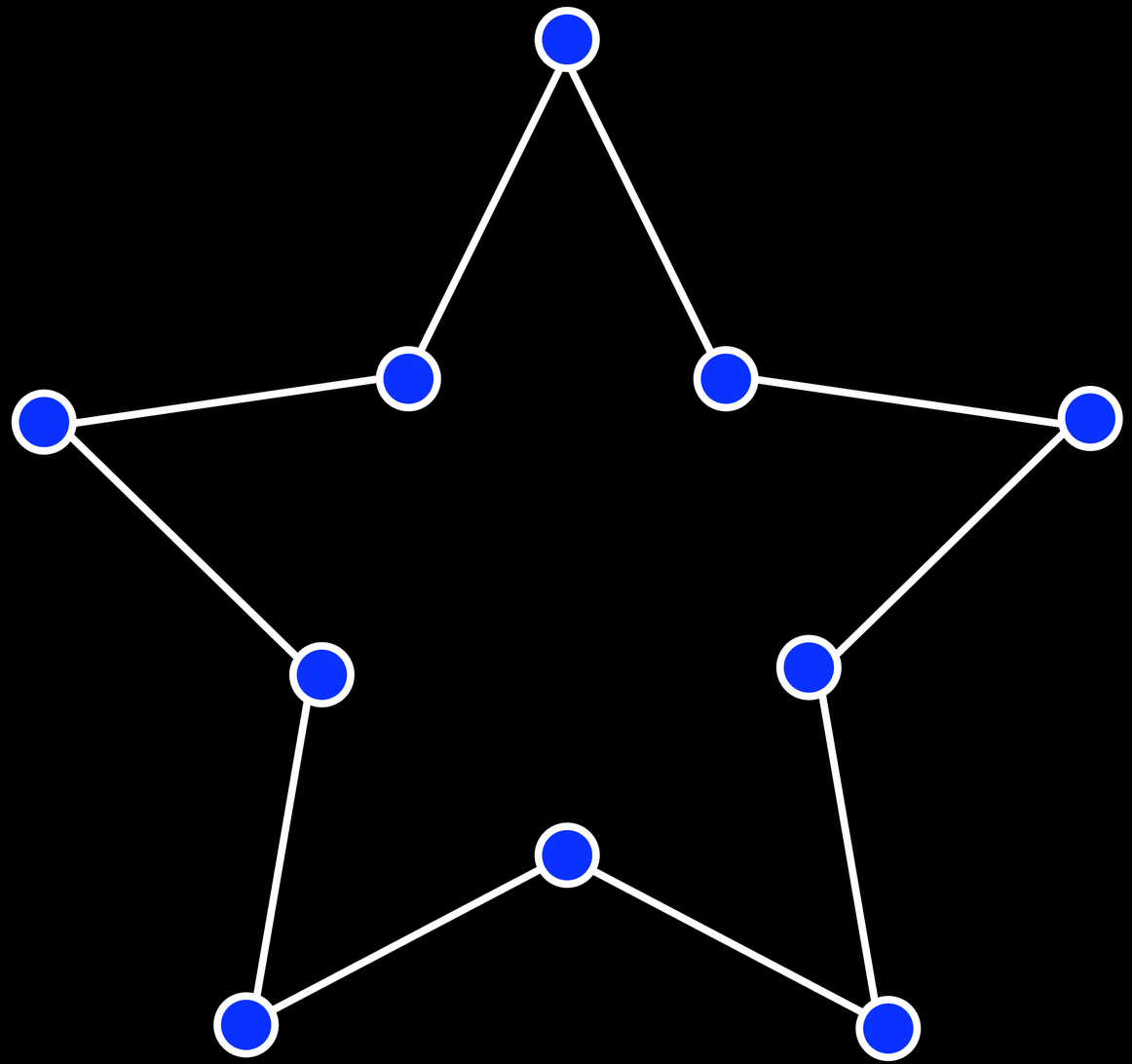
NSGradient

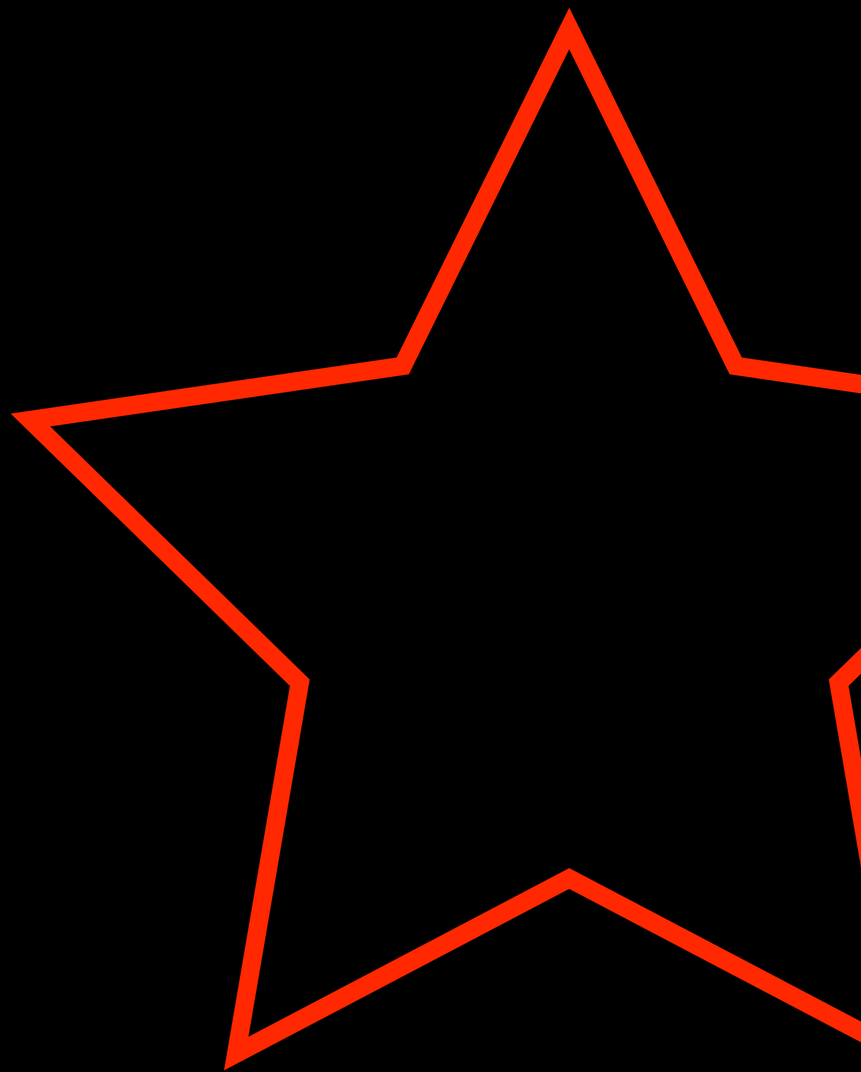


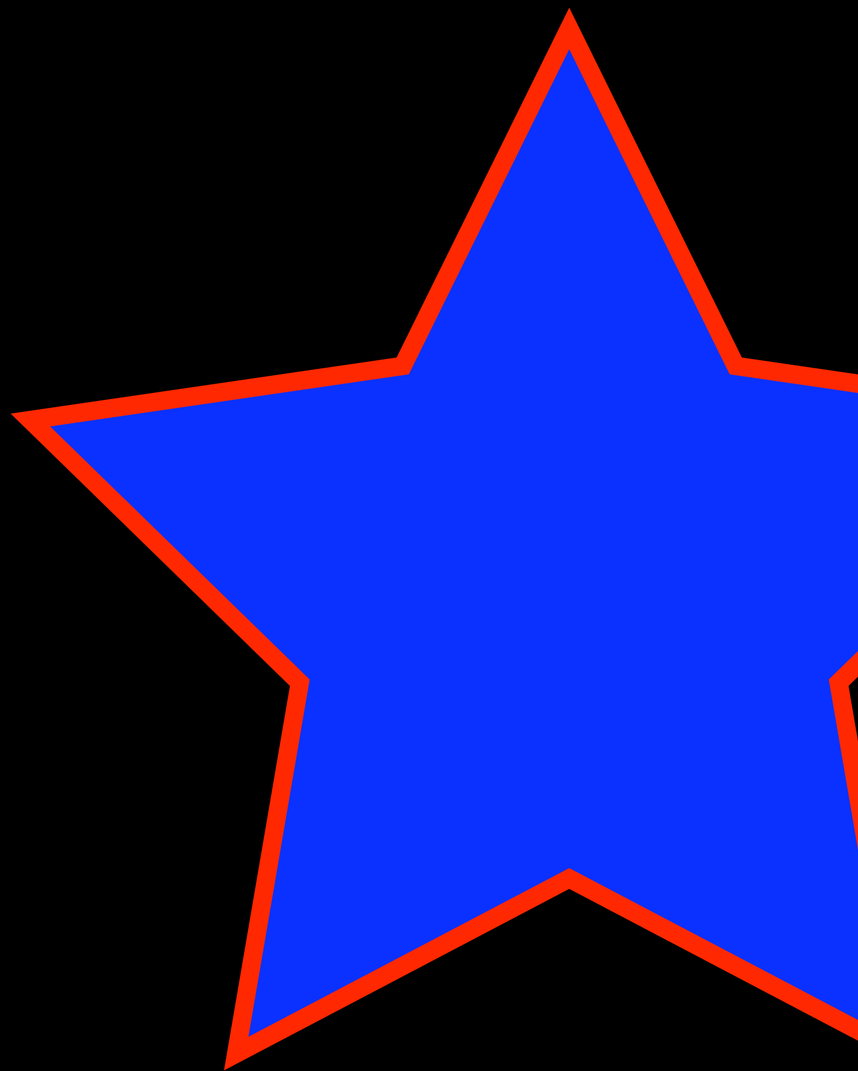
```
NSGradient *gradient;
gradient = [[NSGradient alloc] initWithStartingColor:c1
                                          endingColor:c2];
[gradient drawInRect:frame angle:90.0];
```

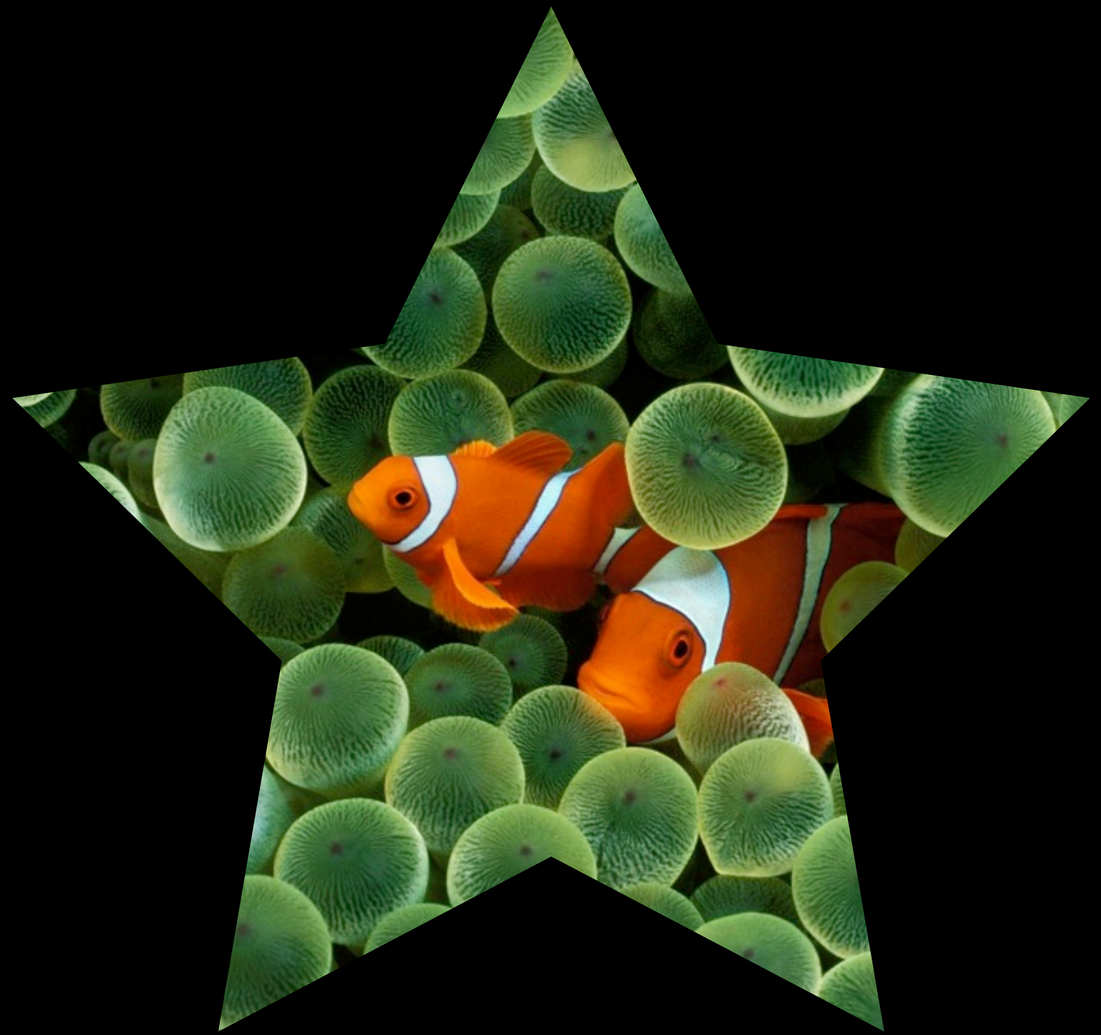
NSBezierPath

```
NSBezierPath *path;
path = [NSBezierPath bezierPathWithRoundedRect:frame
                                       xRadius:48.0
                                       yRadius:48.0];
[path setLineWidth:12.0];
[path stroke];
```

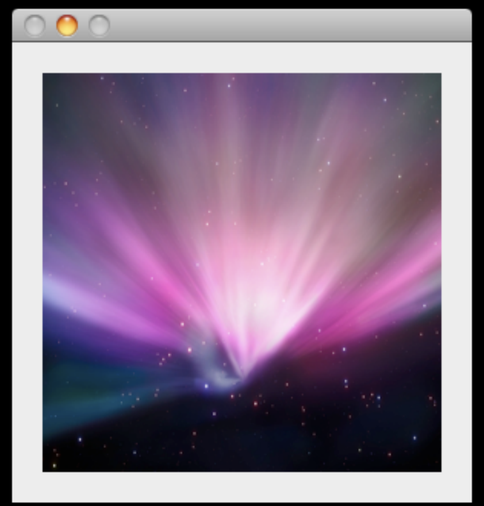






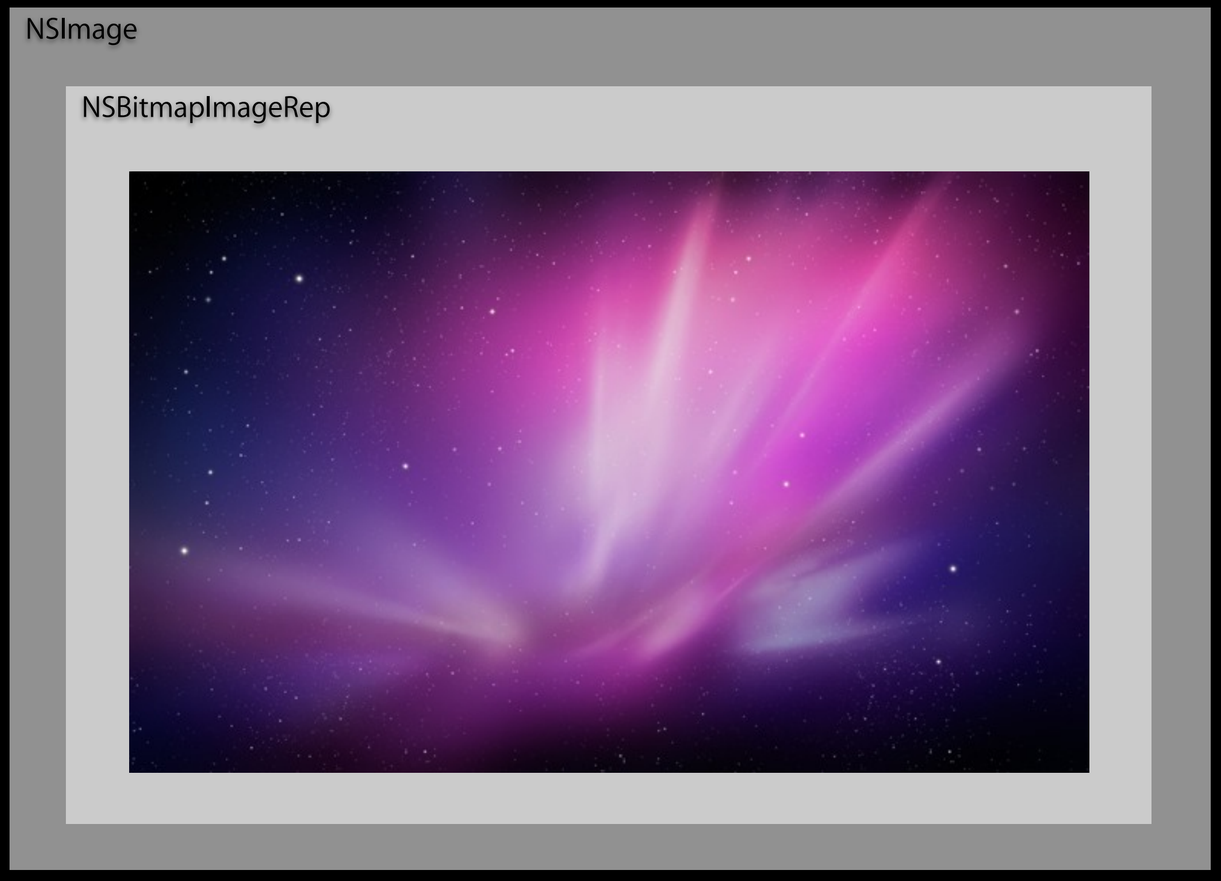


NSImage



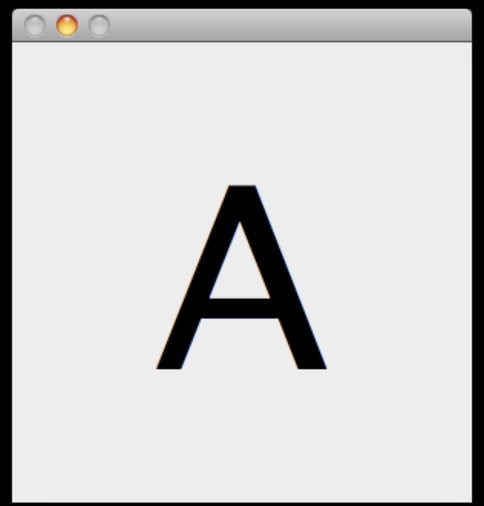
```
NSImage *image = [[NSImage alloc] initWithContentsOfURL:url];
[image drawInRect:cellFrame fromRect:NSZeroRect
        operation:NSCompositeSourceOver fraction:1.0];
```

NSImage vs. NSImageRep



Three and Nine-Part Drawing

NSAttributedString

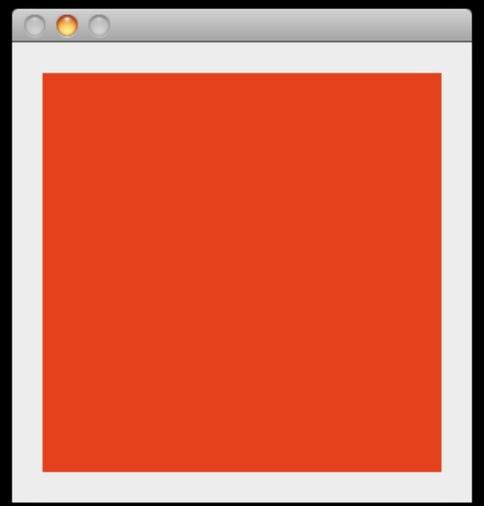



```
NSDictionary *attrs = [[NSDictionary alloc]
initWithObjectsAndKeys:[NSFont systemFontOfSize:165.0],
NSFontAttributeName, nil];
NSAttributedString *aString = [[NSAttributedString alloc]
initWithString:str attributes:attrs];
[aString drawInRect:frame];
```

Object Oriented API

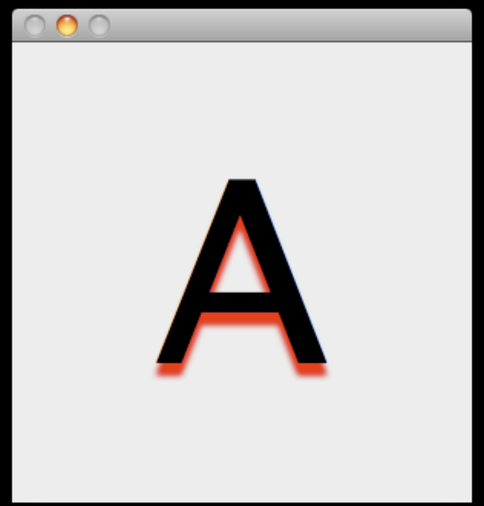
- Colors
- Shadows

NSColor



```
NSColor *color = [NSColor colorWithCalibratedRed:1.0
                                           green:0.0
                                            blue:0.0
                                           alpha:1.0];

[color set];
```

NSShadow




```
NSShadow *shadow = [[NSShadow alloc] init];
[shadow setShadowOffset:NSMakeSize(0.0,-5.0)];
[shadow setShadowColor:[NSColor redColor]];
[shadow set];
```

NSGraphicsContext

- Drawing environment
- Shared configuration
- Save + Restore
- Pre-configured for view drawing

When should you draw?

Event

↓

Change State

↓

Flag Dirty

Callback

↓

Draw

Event

↓

Change State

↓

-setNeedsDisplay:

-drawRect:

↓

Draw

Exposure

PROJECTS

Sample Project 9

Backup 3

Black and white

Gaussian blur

Bloom

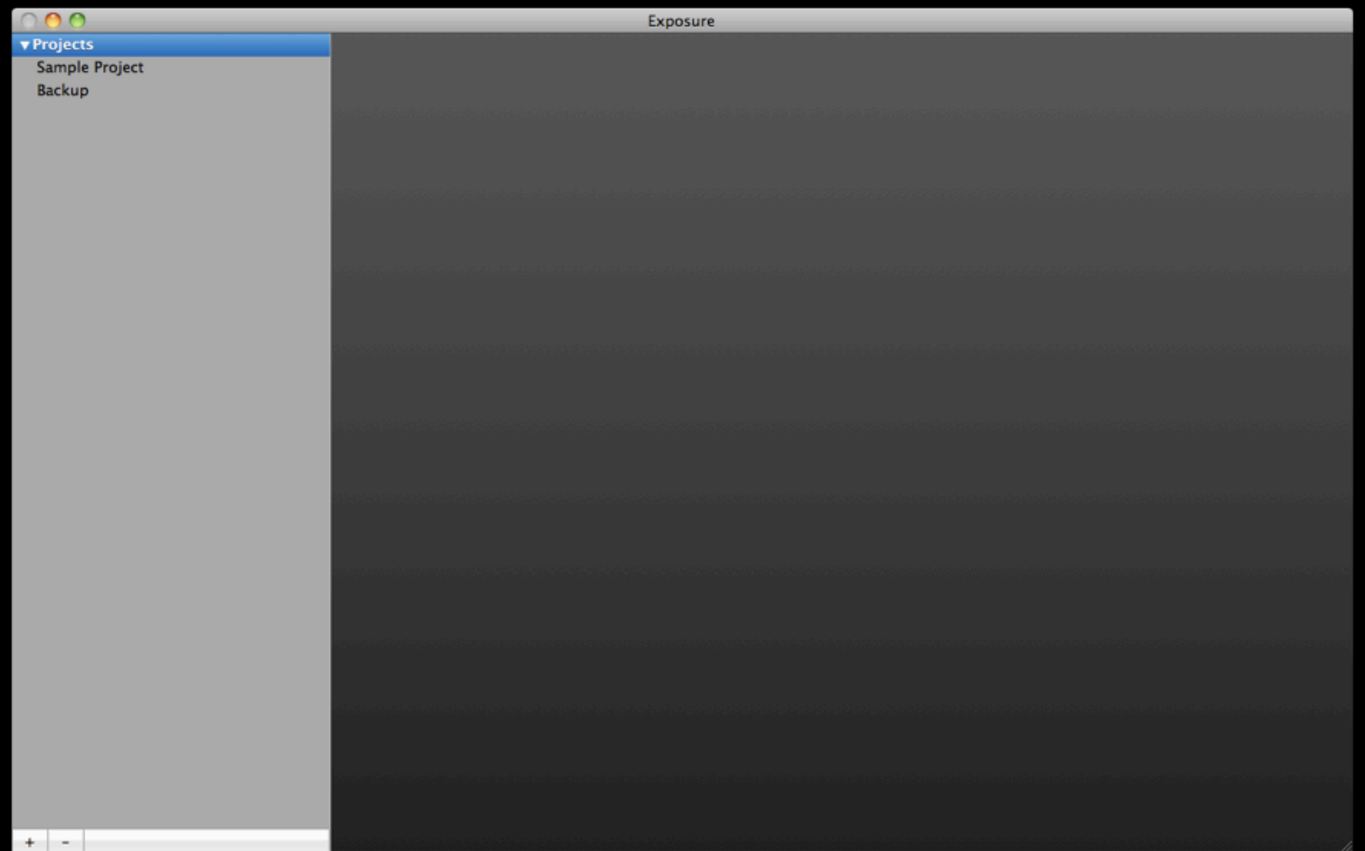
Done

Demo

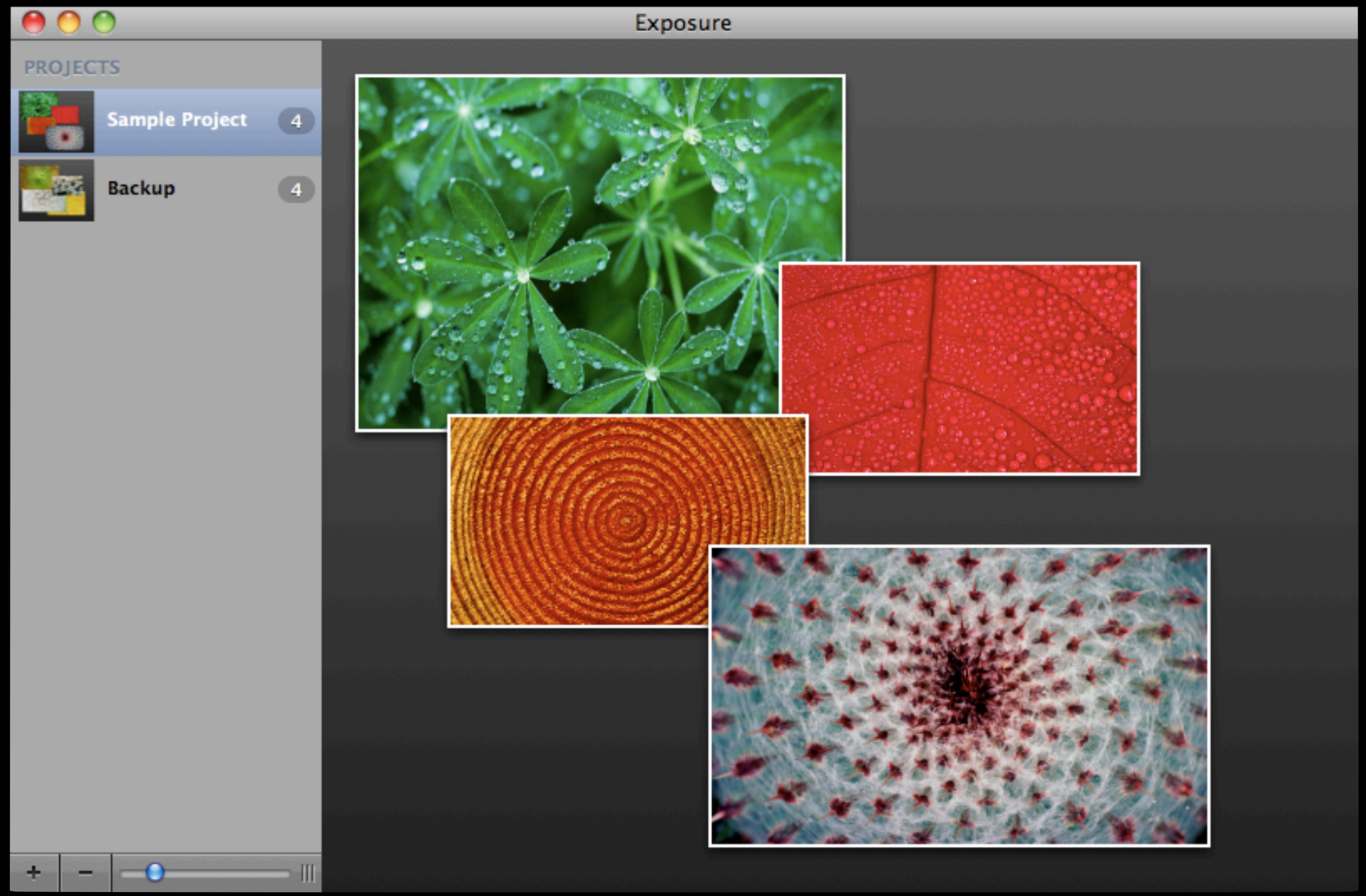
Custom Controls and Behaviours

Ron Lue-Sang
Development Technologies

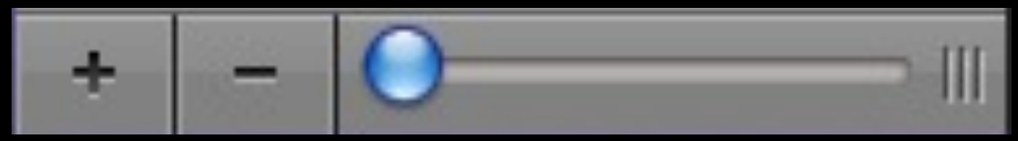
From Start to Finish



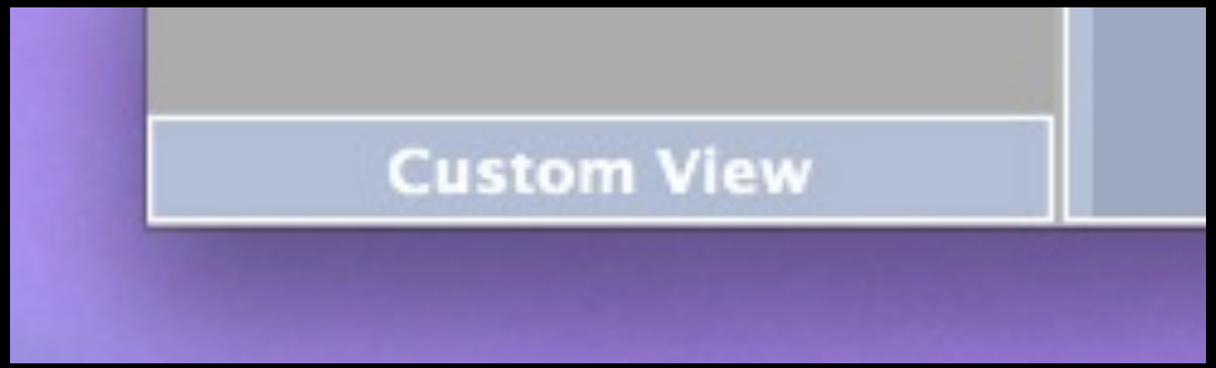
Making Controls Fit the Design

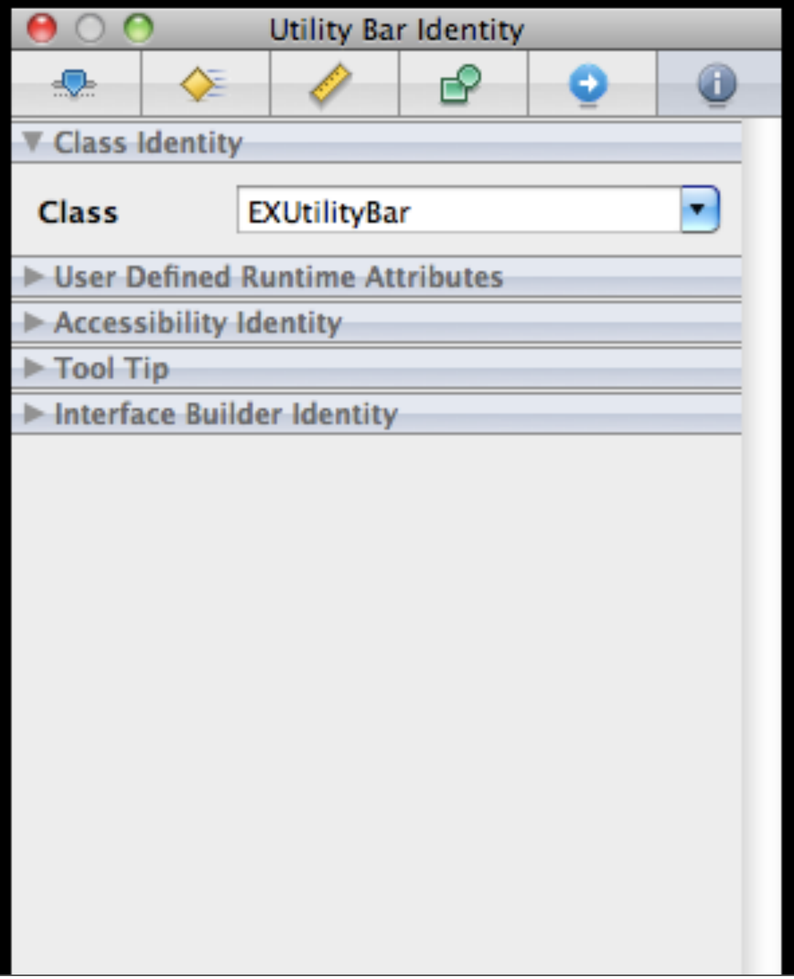


Utility Bar

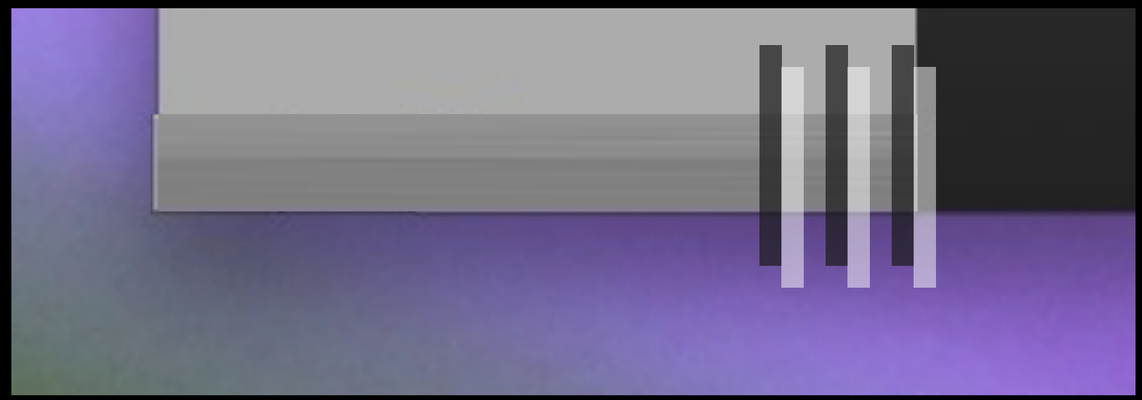


- Gradient button?
- Segmented control?
- Build our own

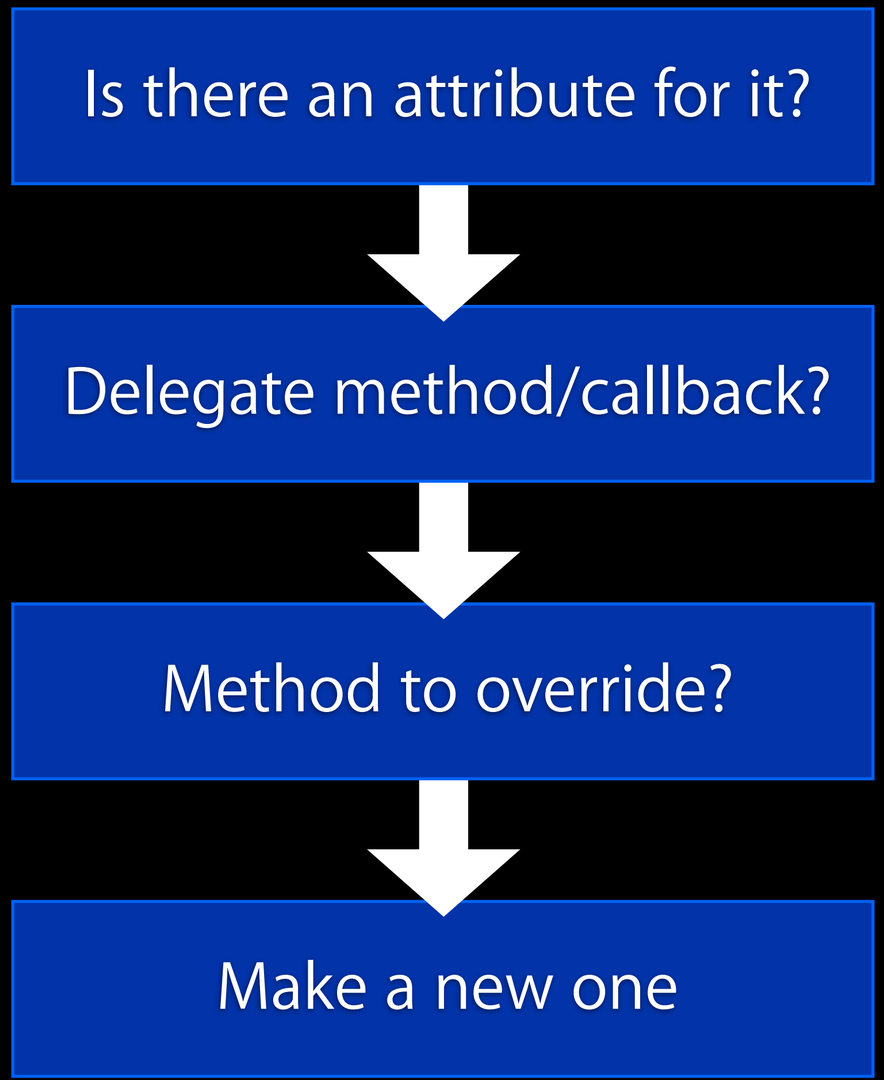

Custom View


Utility Bar Identity
Class Identity
Class
EXUtilityBar
User Defined Runtime Attributes
Accessibility Identity
Tool Tip
Interface Builder Identity

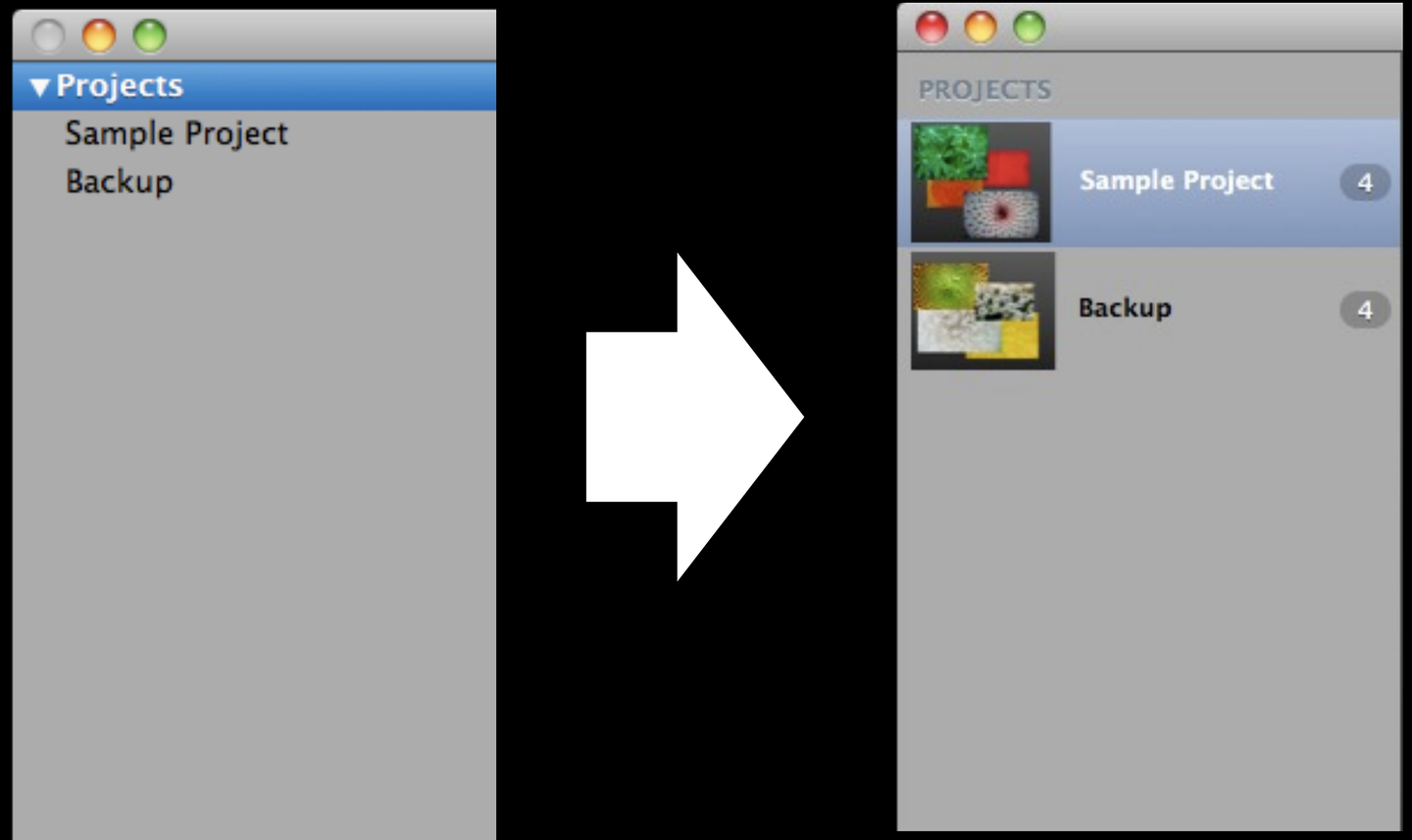
Composition



Making Controls Fit Your Design



Source List

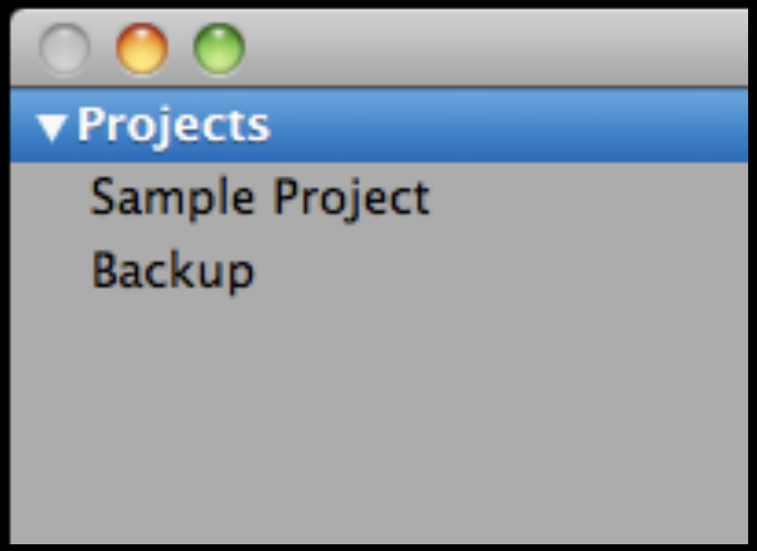

Projects
Sample Project
Backup
PROJECTS
Sample Project
4
Backup
4

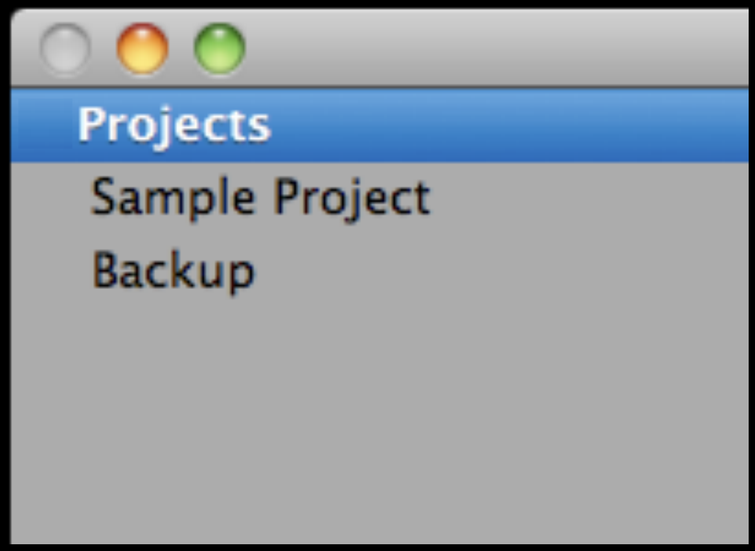
NSOutlineView Delegation

- No Expansion Triangles

```
- (BOOL)outlineView:(NSOutlineView *)outlineView
    shouldShowOutlineCellForItem:(id)item
```

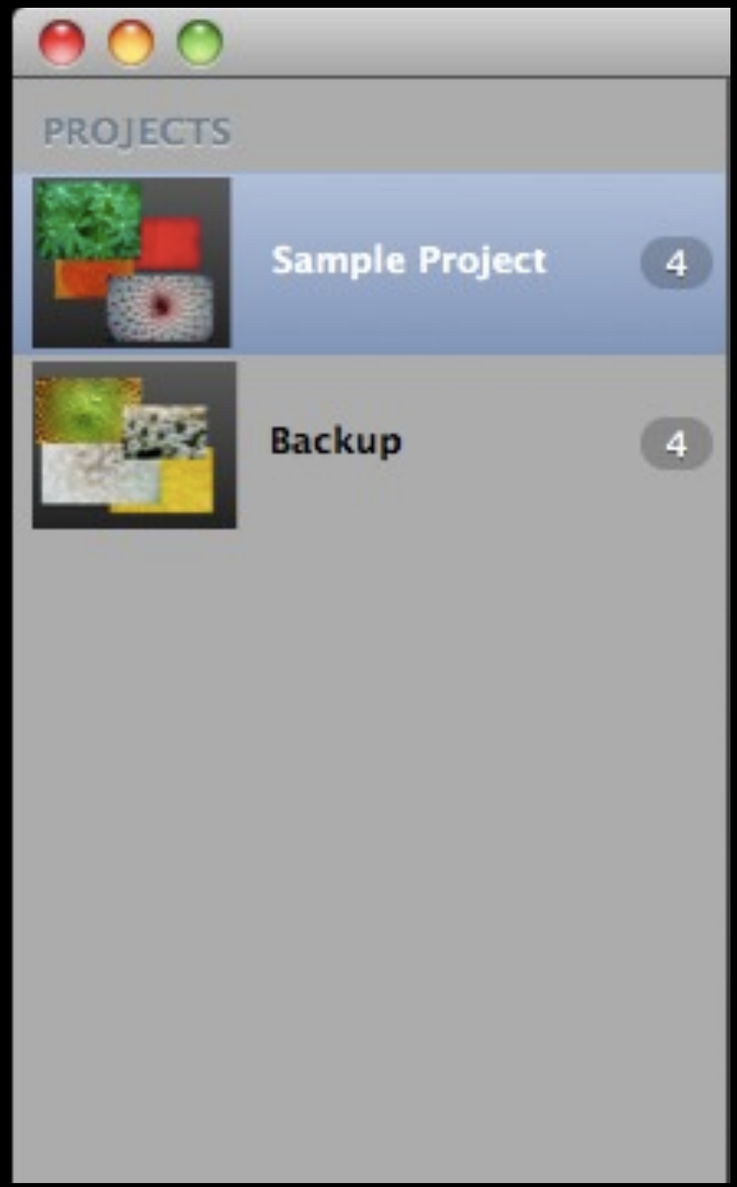
New





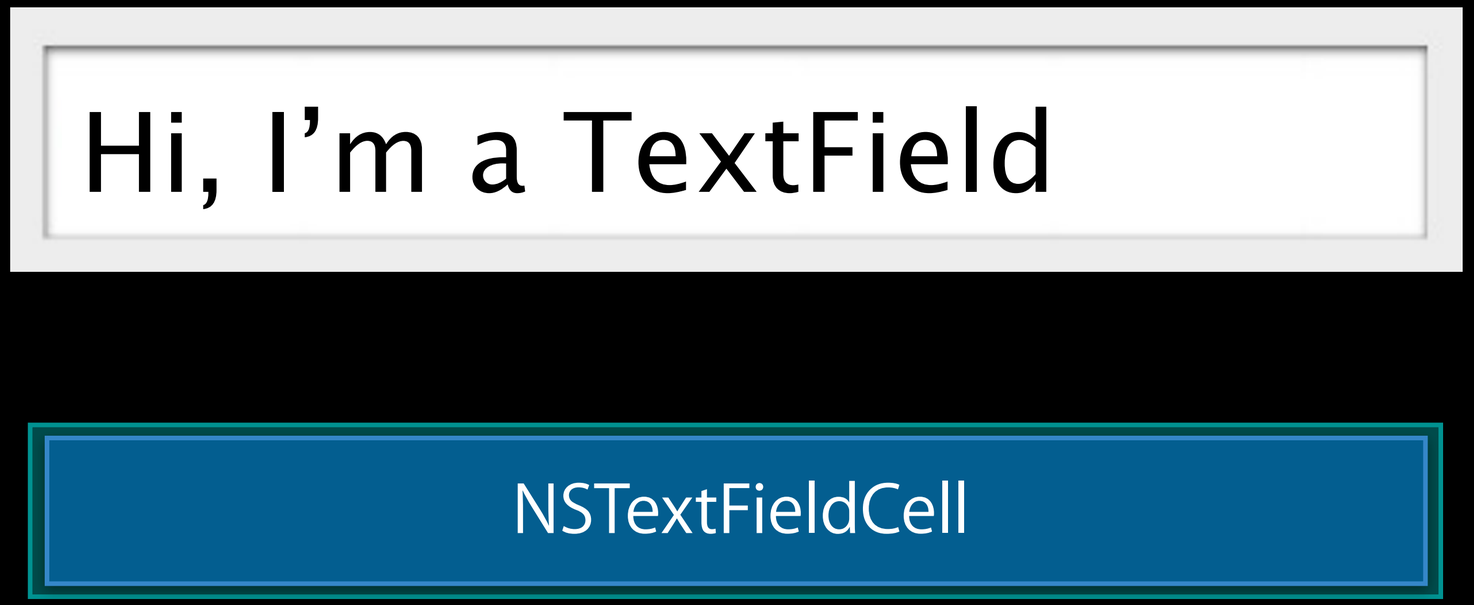
What's Left?

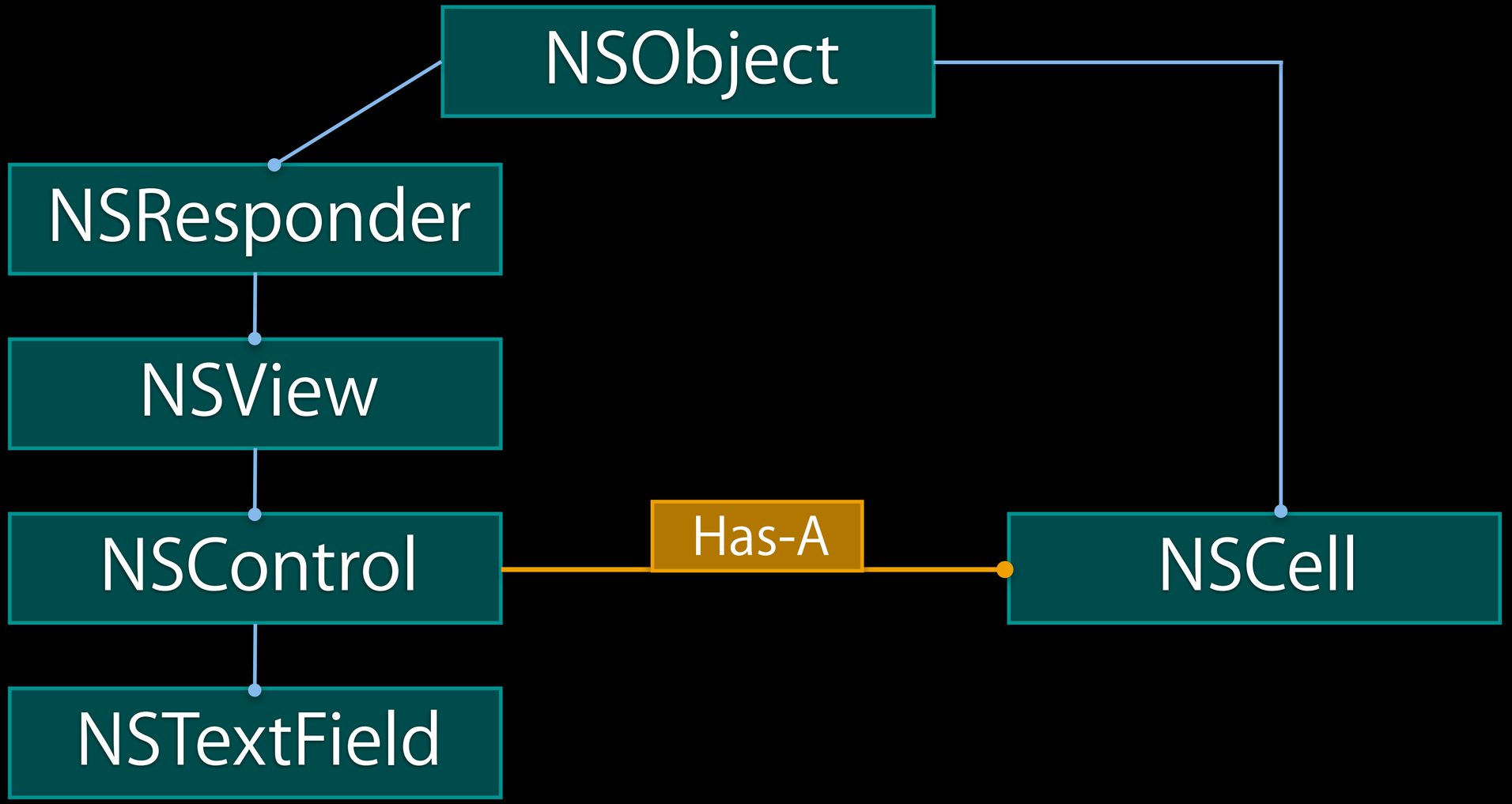
- Icon and text drawing
- Badged rows
- Editing support
- Tooltip placement



Get Back in Your Cell

NSTextField Drawing




NSObject
NSResponder
NSView
NSControl
Has-A
NSCell
NSTextField

NSView

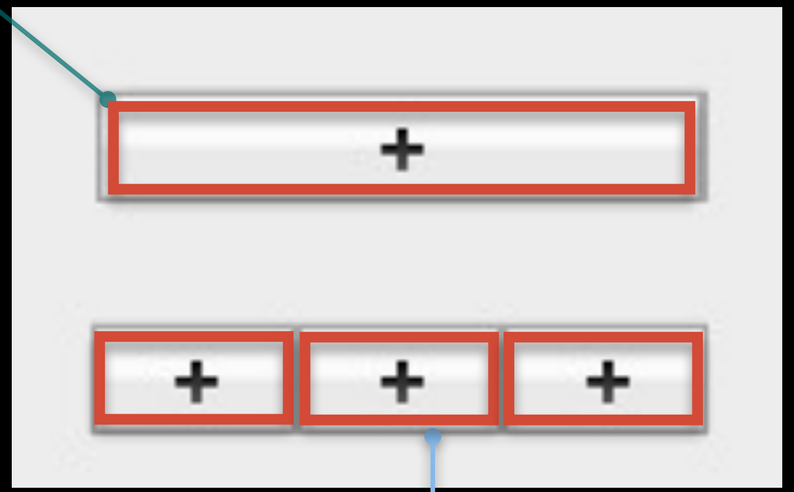
```
- (void)drawRect:(NSRect)rect
```

NSCell

```
- (void)drawWithFrame:(NSRect)cellFrame
               inView:(NSView *)controlView
```

Who Uses NSCell?

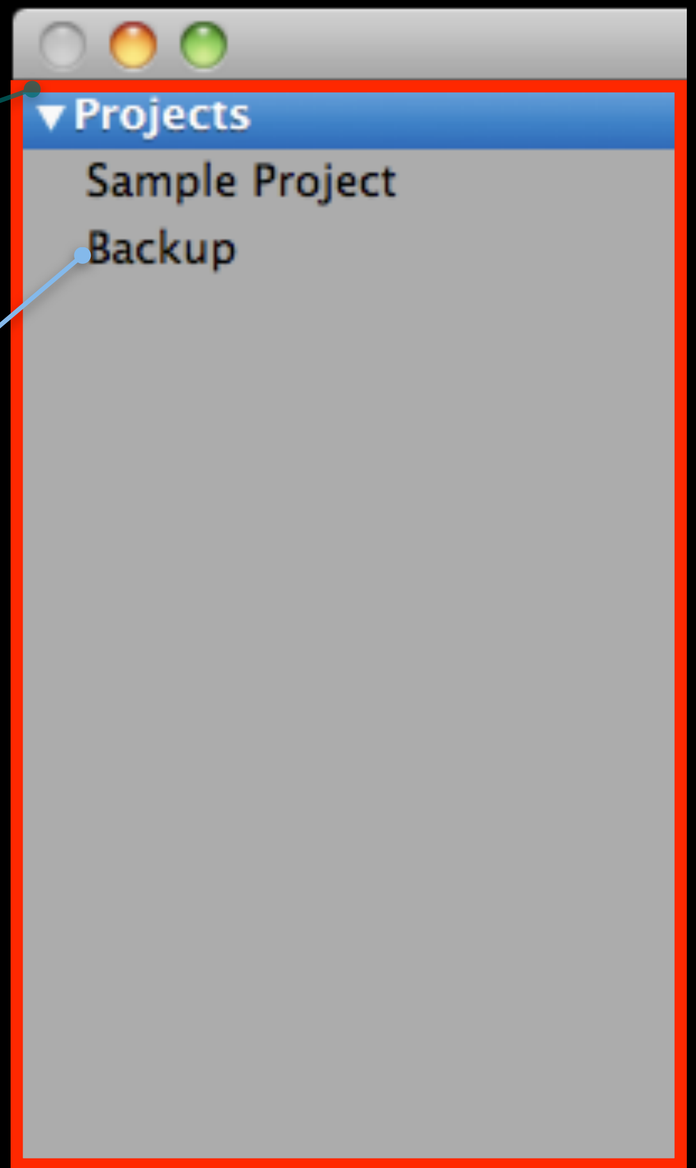
NSButton



NSMatrix

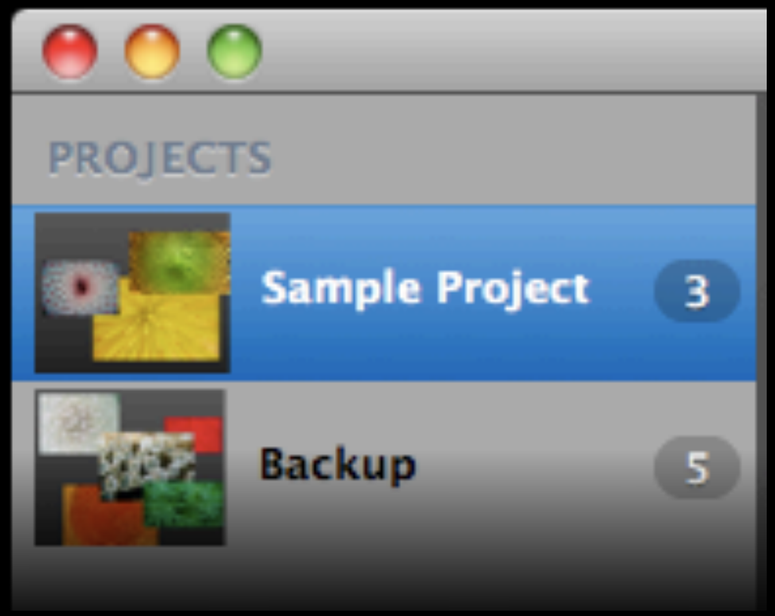
NSOutlineView + Data Cell

NSOutlineView



NSTextFieldCell

NSOutlineView Cell Delegation




```
- (NSCell *)outlineView:(NSOutlineView *)outlineView
 dataCellForTableColumn:(NSTableColumn *)tableColumn
                   item:(id)item
```

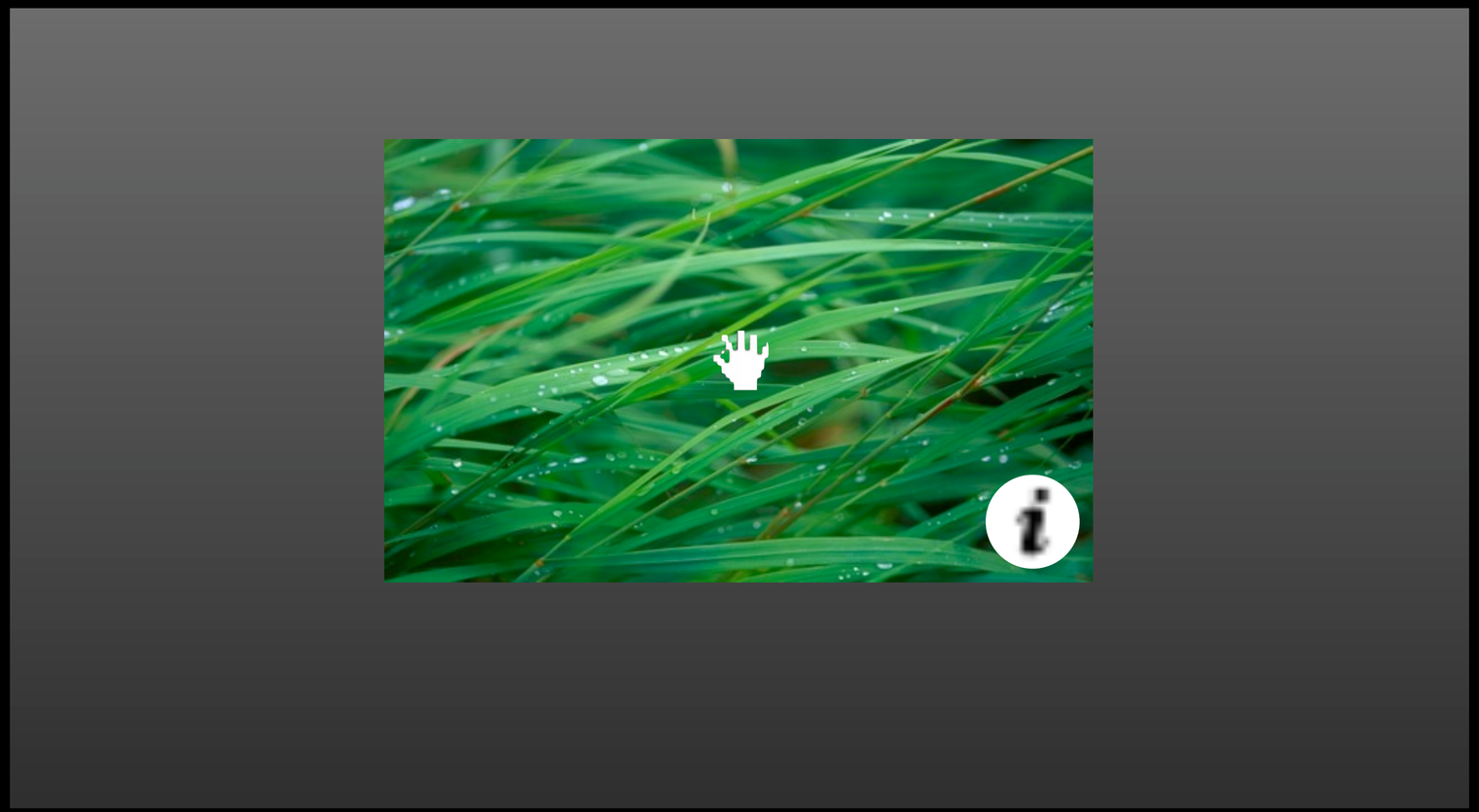
Demo

Behaviour and Animations

Behaviours

- Events
- Drag and drop

```
- (void)mouseEntered:(NSEvent *)theEvent;
- (void)mouseExited:(NSEvent *)theEvent;
```



```
- (void)updateTrackingAreas;
```

```
- (void)removeTrackingArea:(NSTrackingArea *)trackingArea;
- (void)addTrackingArea:(NSTrackingArea *)trackingArea;
```

NSTrackingArea

```
- (id)initWithRect:(NSRect)rect
           options:(NSTrackingAreaOptions)options
             owner:(id)owner
          userInfo:(NSDictionary *)userInfo

 // NSTrackingMouseEnteredAndExited
```

```
- (void)mouseEntered:(NSEvent *)theEvent;
- (void)mouseExited:(NSEvent *)theEvent;
```

NSResponder and Events

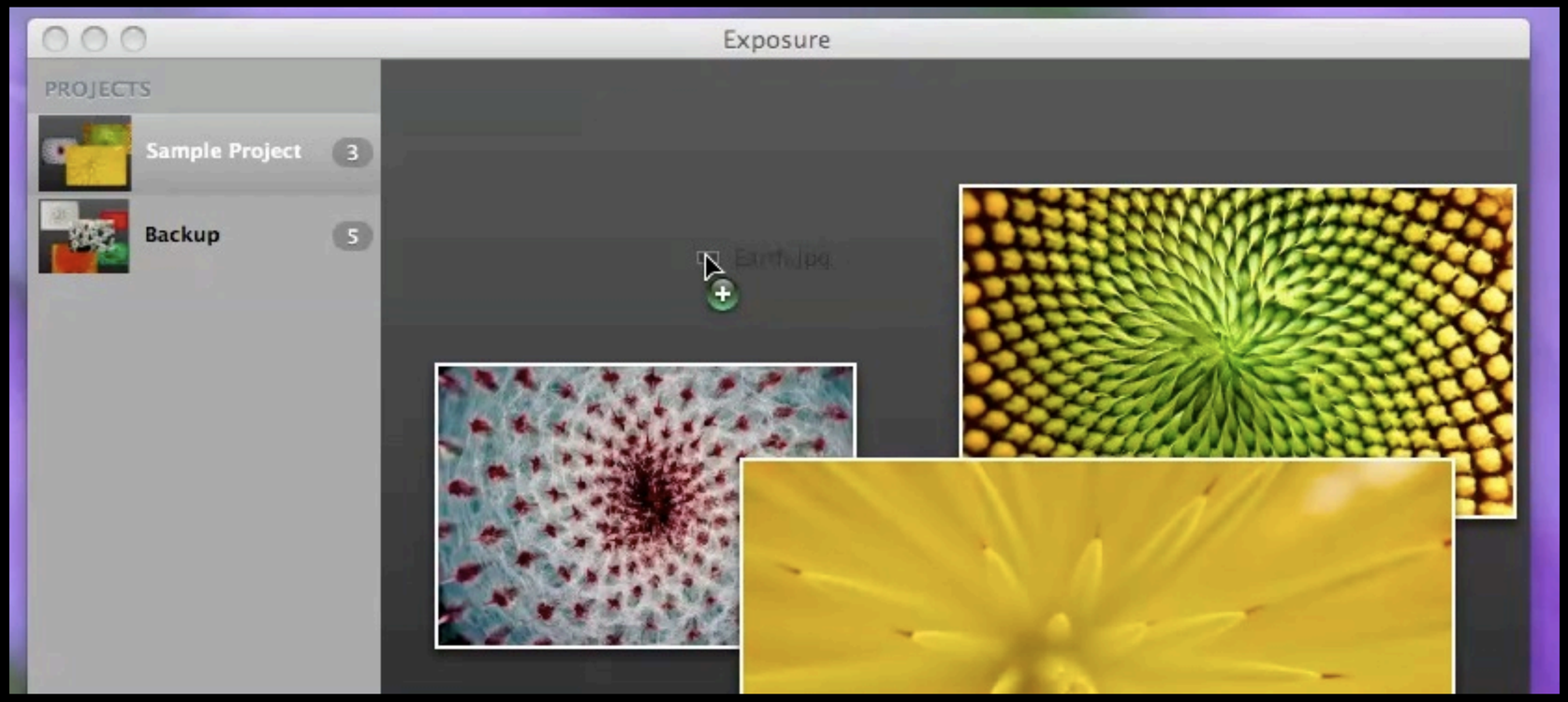
```
- (void)mouseDown:(NSEvent *)theEvent;
- (void)mouseUp:(NSEvent *)theEvent;
- (void)mouseDragged:(NSEvent *)theEvent;
- (void)mouseEntered:(NSEvent *)theEvent;
- (void)mouseExited:(NSEvent *)theEvent;
```

```
- (void)mouseDown:(NSEvent *)theEvent {

    // Get mouse position from event
   NSPoint location = [theEvent locationInWindow];

  // Get the location in our own bounds
    location = [self convertPoint:location
                         fromView:nil];


}
```

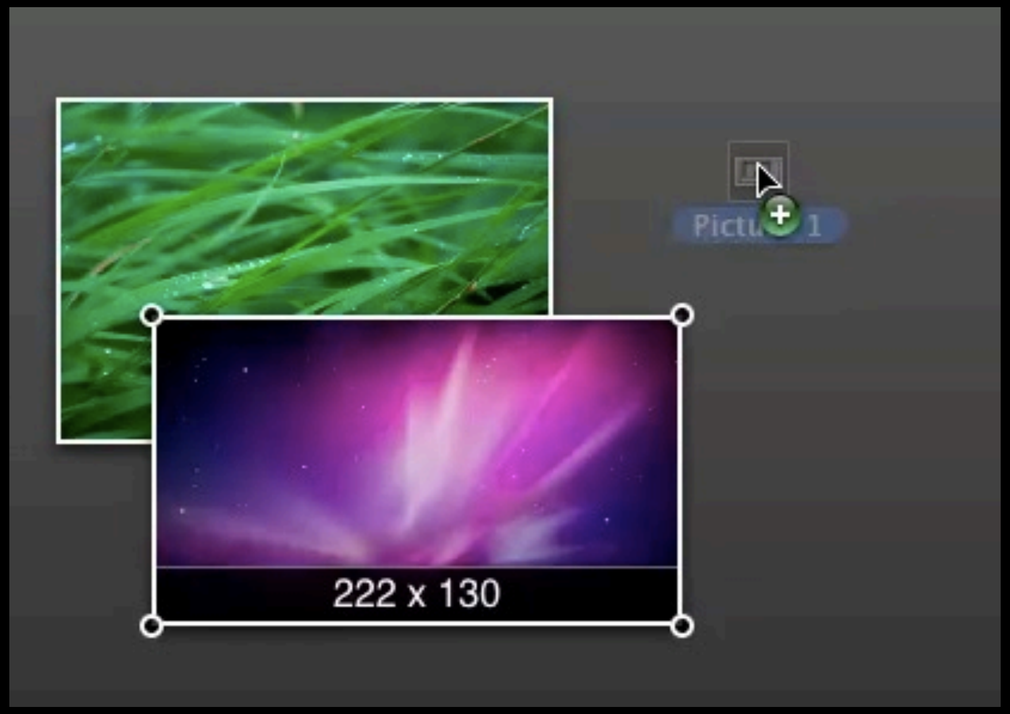
```
- (void)mouseDown:(NSEvent *)theEvent {

    // Get mouse position from event
   NSPoint location = [theEvent locationInWindow];

  // Get the location in our own bounds
    location = [self convertPoint:location
                         fromView:nil];

// Keep track of what was under the mouse
}
```

```
- (void)mouseDragged:(NSEvent *)event {

// What was under mouse when mouseDown started?

    [self trackMouseAndMoveTile];
}
```

Drag and Drop



MouseDragged Events != Drag and Drop



```
- (void)registerForDraggedTypes:(NSArray *)pboardTypes
    // awakeFromNib timeframe


- (BOOL)performDragOperation:(id <NSDraggingInfo>)sender
    // Interrogate sender for data
```

draggingSource
draggingLocation
draggingPasteboard

Animations

Normal View Drawing

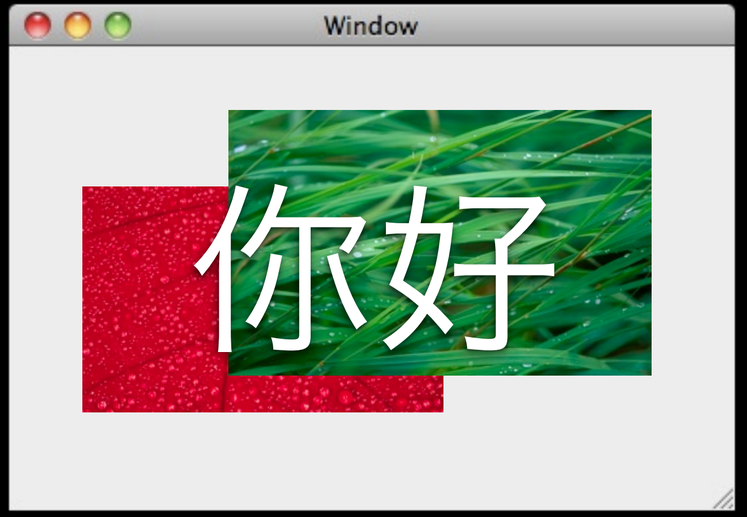
Hello

Normal View Drawing

你好

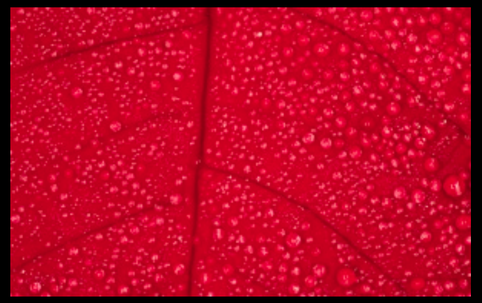
Normal View Drawing

你好



Layer Backed Views

```
- (void)setWantsLayer:(BOOL)wantsLayer;
```



你好

CoreAnimation

- CALayer
- CAAnimation
- NSAnimatablePropertyContainer

Layer Backed Views

```
for (NSView *view in viewsToMove) {

  // MOVES view to new location and size
  [view setFrame:frame];

  // ANIMATES view to new location and size
  [[view animator] setFrame:frame];

}
```

Layer Backed Views

```
for (NSView *view in viewsToMove) {

  // Animate frame and opacity change
  [[view animator] setFrame:frame];
  [[view animator] setAlphaValue:0.0];

}
```

Demo

Related Sessions

User Events in Cocoa	Nob Hill Wednesday 3:30PM
Getting Started with Interface Builder	Pacific Heights Wednesday 9:00AM
Core Animation Techniques for iPhone and Mac	Mission Wednesday 9:00AM
Presenting User Data with Table Views and Browsers	Nob Hill Thursday 10:30AM
NSImage in Snow Leopard	Nob Hill Thursday 3:30PM

Labs

Cocoa Open Lab	OS X Lab C Tuesday 3:30PM
Quartz 2D Lab	Graphics and Media Lab A Wednesday 12:00PM
Core Animation Lab	Graphics and Media Lab B Wednesday 12:00PM
Getting Started with Interface Builder Lab	iPhone Lab A Wednesday 2:00PM
Cocoa Open Lab	OS X Lab C Thursday 2:00PM
Mac OS X Interface Design Consulting Lab	Interface Design Lab Every Afternoon

More Information

Michael Jurewitz
Developer Tools Evangelist
jurewitz@apple.com

Documentation

Cocoa Drawing Guide
http://developer.apple.com/documentation/Cocoa/Conceptual/CocoaDrawingGuide

View Programming Guide for Cocoa
http://developer.apple.com/documentation/Cocoa/Conceptual/CocoaViewsGuide

Apple Human Interface Guidelines
http://developer.apple.com/documentation/userexperience/Conceptual/AppleHIGuidelines

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09